

18 JAN 2005

PCT/JP03/09153

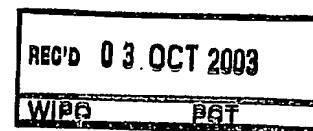
日本国特許庁  
JAPAN PATENT OFFICE

15.08.03

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日  
Date of Application: 2002年 7月19日



出願番号  
Application Number: 特願 2002-211021

[ST. 10/C]: [JP 2002-211021]

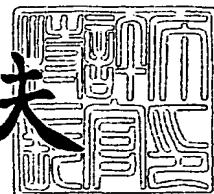
出願人  
Applicant(s): 独立行政法人産業技術総合研究所

PRIORITY DOCUMENT  
SUBMITTED OR TRANSMITTED IN  
COMPLIANCE WITH  
RULE 17.1(a) OR (b)

2003年 9月19日

特許庁長官  
Commissioner,  
Japan Patent Office

今井 康夫



出証番号 出証特 2003-3076955

Best Available Copy

【書類名】 特許願  
【整理番号】 113MS0348  
【提出日】 平成14年 7月19日  
【あて先】 特許庁長官殿  
【国際特許分類】 G06F 17/00  
【発明者】  
【住所又は居所】 兵庫県尼崎市若王寺3丁目11番46号 独立行政法人  
産業技術総合研究所関西センター尼崎事業所内  
【氏名】 大崎 人士  
【発明者】  
【住所又は居所】 兵庫県尼崎市若王寺3丁目11番46号 独立行政法人  
産業技術総合研究所関西センター尼崎事業所内  
【氏名】 高井 利憲  
【特許出願人】  
【識別番号】 301021533  
【氏名又は名称】 独立行政法人産業技術総合研究所  
【代表者】 理事長 吉川 弘之  
【連絡先】 0727-51-9681  
【提出物件の目録】  
【物件名】 明細書 1  
【物件名】 図面 1  
【物件名】 要約書 1  
【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 リアクティブ・システムの安全性検証装置、方法、プログラム及びそのプログラムを記録した記録媒体

【特許請求の範囲】

【請求項 1】 関数記号の集合、書換規則の集合、公理の集合、項の集合、及び検査対象となる項の集合によって表わされるリアクティブ・システムの安全性検証装置であって、

前記公理の集合が、交換則及び結合則のみを要素とする集合であり、

前記公理の集合の下で、前記項の集合を受理する第1の等式付ツリー・オートマトンを生成する翻訳部、

前記第1の等式付ツリー・オートマトンを初期データとして、前記書換規則の集合及び前記公理の集合の下で、前記項の集合から派生する項からなる集合と前記項の集合とを受理する第2の等式付ツリー・オートマトンを生成するシミュレーション部、及び

前記第2の等式付ツリー・オートマトンと、前記検査対象となる項の集合を受理する第3の等式付ツリー・オートマトンとを結合して第4の等式付ツリー・オートマトンを生成し、該第4の等式付ツリー・オートマトンが受理する集合が空集合か否かを判断する集合演算部を備えていることを特徴とするリアクティブ・システムの安全性検証装置。

【請求項 2】 関数記号の集合、書換規則の集合、公理の集合、項の集合、及び検査対象となる項の集合によって表わされるリアクティブ・システムの安全性検証装置であって、

前記公理の集合が、交換則及び結合則のみを要素とする集合であり、

前記公理の集合の下で、前記項の集合を受理する第1の等式付ツリー・オートマトンを生成する翻訳部、

前記第1の等式付ツリー・オートマトンを初期データとして、前記書換規則の集合及び前記公理の集合の下で、前記項の集合から派生する項からなる集合と前記項の集合とを受理する第2の等式付ツリー・オートマトンを生成するシミュレーション部、及び

前記第2の等式付ツリー・オートマトンが、前記検査対象となる項を受理するか否かを判断する集合演算部を備えていることを特徴とするリアクティブ・システムの安全性検証装置。

【請求項3】 前記関数記号の集合は、暗号処理、復号処理、及び通信処理を表す関数を要素として含む集合であり、

前記書換規則の集合は、暗号処理された情報が復号処理されることによって平文に戻ることを表す規則を要素として含む集合であり、

前記検査対象となる項は、秘密情報であり、

前記項の集合は、秘密情報を交換する複数の主体の各々の知識の集合、及びこれら複数の主体間で交換される情報を傍受する主体の知識の集合であることを特徴とする請求項1又は2に記載のリアクティブ・システムの安全性検証装置。

【請求項4】 関数記号の集合、書換規則の集合、公理の集合、項の集合、及び検査対象となる項の集合によって表わされるリアクティブ・システムの安全性検証方法であって、

前記公理の集合が、交換則及び結合則のみを要素とする集合であり、

前記公理の集合の下で、前記項の集合を受理する第1の等式付ツリー・オートマトンを生成する第1のステップ、

前記第1の等式付ツリー・オートマトンを初期データとして、前記書換規則の集合及び前記公理の集合の下で、前記項の集合から派生する項からなる集合と前記項の集合とを受理する第2の等式付ツリー・オートマトンを生成する第2のステップ、及び

前記第2の等式付ツリー・オートマトンと、前記検査対象となる項の集合を受理する第3の等式付ツリー・オートマトンとを結合して第4の等式付ツリー・オートマトンを生成し、該第4の等式付ツリー・オートマトンが受理する集合が空集合か否かを判断する第3のステップを含むことを特徴とするリアクティブ・システムの安全性検証方法。

【請求項5】 関数記号の集合、書換規則の集合、公理の集合、項の集合、及び検査対象となる項の集合によって表わされるリアクティブ・システムの安全性検証方法であって、

前記公理の集合が、交換則及び結合則のみを要素とする集合であり、

前記公理の集合の下で、前記項の集合を受理する第1の等式付ツリー・オートマトンを生成する第1のステップ、

前記第1の等式付ツリー・オートマトンを初期データとして、前記書換規則の集合及び前記公理の集合の下で、前記項の集合から派生する項からなる集合と前記項の集合とを受理する第2の等式付ツリー・オートマトンを生成する第2のステップ、及び

前記第2の等式付ツリー・オートマトンが、前記検査対象となる項を受理するか否かを判断する第3のステップを含むことを特徴とするリアクティブ・システムの安全性検証方法。

**【請求項6】** 前記関数記号の集合は、暗号処理、復号処理、及び通信処理を表す関数を要素として含む集合であり、

前記書換規則の集合は、暗号処理された情報が復号処理されることによって平文に戻ることを表す規則を要素として含む集合であり、

前記検査対象となる項は、秘密情報であり、

前記項の集合は、秘密情報を交換する複数の主体の各々の知識の集合、及びこれら複数の主体間で交換される情報を傍受する主体の知識の集合であることを特徴とする請求項4又は5に記載のリアクティブ・システムの安全性検証方法。

**【請求項7】** コンピュータに、

関数記号の集合、書換規則の集合、公理の集合、項の集合、及び検査対象となる項の集合によって表わされた手順の入力を受け付ける機能、

交換則及び結合則のみを要素とする前記公理の集合の下で、前記項の集合を受理する第1の等式付ツリー・オートマトンを生成する機能、

前記第1の等式付ツリー・オートマトンを初期データとして、前記書換規則の集合及び前記公理の集合の下で、前記項の集合から派生する項からなる集合と前記項の集合とを受理する第2の等式付ツリー・オートマトンを生成する機能、及び

前記第2の等式付ツリー・オートマトンと、前記検査対象となる項の集合を受理する第3の等式付ツリー・オートマトンとを結合して第4の等式付ツリー・オ

ートマトンを生成し、該第4の等式付ツリー・オートマトンが受理する集合が空集合か否かを判断する機能を実現させるためのリアクティブ・システムの安全性検証プログラム。

【請求項8】 コンピュータに、

関数記号の集合、書換規則の集合、公理の集合、項の集合、及び検査対象となる項の集合によって表わされた手順の入力を受け付ける機能、

交換則及び結合則のみを要素とする前記公理の集合の下で、前記項の集合を受理する第1の等式付ツリー・オートマトンを生成する機能、

前記第1の等式付ツリー・オートマトンを初期データとして、前記書換規則の集合及び前記公理の集合の下で、前記項の集合から派生する項からなる集合と前記項の集合とを受理する第2の等式付ツリー・オートマトンを生成する機能、及び

前記第2の等式付ツリー・オートマトンが、前記検査対象となる項を受理するか否かを判断する機能を実現させるためのリアクティブ・システムの安全性検証プログラム。

【請求項9】 前記関数記号の集合は、暗号処理、復号処理、及び通信処理を表す関数を要素として含む集合であり、

前記書換規則の集合は、暗号処理された情報が復号処理されることによって平文に戻ることを表す規則を要素として含む集合であり、

前記検査対象となる項は、秘密情報であり、

前記項の集合が、秘密情報を交換する複数の主体の各々の知識の集合、及びこれら複数の主体間で交換される情報を傍受する主体の知識の集合であることを特徴とする請求項7又は8に記載のリアクティブ・システムの安全性検証プログラム。

【請求項10】 請求項7～9の何れかの項に記載のプログラムを記録したコンピュータ読取可能な記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、ツリー・オートマトン理論に基づきリアクティブ・システムの安全性を検証する装置、方法、プログラム及びそのプログラムを記録した記録媒体に関する。

### 【0002】

#### 【従来の技術】

近年、コンピュータ技術、通信技術の進歩に伴い、様々な産業分野において、公衆ネットワーク及び専用ネットワークを介して大量の情報交換が行われており、秘密情報の交換が必要となる場合が多く発生している。特に、金融ビジネス、電子商取引などの分野においては、通信者間の認証、通信情報の機密保持などが高い精度で要求されており、暗号処理などの安全性を確保するための各種方法が開発され、利用されている。提供されるサービスの安全性は、暗号の堅牢性が保証されている場合には、主として使用される暗号通信手順の安全性（例えば、暗号化情報を復号化処理する場合に用いる「秘密鍵」が、意図する相手にのみ受け渡しがされていることの確実性）に依存することとなる。従って、開発された暗号通信手順の安全性の検証が、秘密情報の秘匿性を保証する上で、非常に重要な技術となる。

### 【0003】

本願において、「暗号通信手順」には、データを暗号処理、復号処理する手順、及び暗号化されたデータが通信回線を介して交換される手順が含まれるが、通信規格が定めるデータグラムのビット形式や通信経路の動的制御などの実装の方法は含まれない。

### 【0004】

また、「安全性検証」とは、暗号通信手順に限らず、動作中に外界からの刺激を受けて、その刺激に対する応答を返す動作を繰り返すシステムであるリアクティブ・システムの動作手順を対象として、それが意図通りに記述されているか否かを確認することを指し、システムが如何なる場合にも意図しない状態、例えば危険な状態にならないとき、そのシステムが「安全」であるという。暗号情報を通信するシステムもリアクティブ・システムの一種であり、暗号通信手順をシステムの動作手順とみなすことができる。暗号通信手順の安全性検証においては、

実際の通信回線の電気的な信頼性及び品質は、検証の対象外であり、安全性とは交換される情報の秘密保持性を意味する。

#### 【0005】

従来、暗号通信手順の検証方法として、オートマトン理論に基づく正則ツリー・オートマトンと呼ばれる枠組みを用いた方法が知られている。初期に提案された検証方法は、楫勇一、藤原融、嵩忠雄による “Solving a Unification Problem under Constrained Substitutions Using Tree Automata(ツリー・オートマトンを用いた制約付代入における单一化問題の解法)” (Journal of Symbolic Computation 23(1), pp.79-117, 1997) に開示されている。

#### 【0006】

上記の方法を発展させた方法が、David Monniauxによる “Abstracting Cryptographic Protocols with Tree Automata(ツリー・オートマトンによる暗号通信プロトコルの抽出化法)” (Proceeding of 6th International Static Analysis Symposium, Venice(Italy), Lecture Notes in Computer Science 1694, pp.149-163, 1999) 、及びThomas Genet, Francis Klayによる “Rewriting for Cryptographic Protocol Verification(暗号通信プロトコル検証のための書き換系)” (Proceeding of 17th International Conference on Automated Deduction, Pittsburgh(PA), Lecture Notes in Computer Science 1831, pp.271-290, 2000) に開示されている。

#### 【0007】

オートマトンとは、実際の装置、システムなどを抽象的に表現した系であり、複数の状態を取ることができ、「入力」によって各状態間の遷移が起こる。取り得る状態は、必ずしも有限とは限らない。1つ又は一連の複数の入力 INPUT によって、オートマトンが、初期状態から所定の終了状態に至った場合、INPUT がオートマトンによって受理されたという。一般に、オートマトンは  $(\Sigma, Q, Q_f, \Delta)$  と記述される。ここで、 $\Sigma$  は入力の集合、 $Q$  は取り得る状態の集合、 $Q_f$  は終了状態の集合、 $\Delta$  は遷移規則の集合である。

#### 【0008】

従って、ある集合の構成要素だけを受理し、その他を受理しないオートマトン

を論理式として記述することができれば、その集合に対する処理、即ちその集合の要素に対する処理を、その論理式を用いて等価的に行うことができる。このことは、処理対象の集合が無限の要素からなる場合に、特に有効である。

#### 【0009】

ツリー・オートマトンとは、ツリー構造を有するデータを受理するオートマトンを表す。また、正則ツリー・オートマトンとは、正則性の条件を満たすツリー・オートマトンを表す。

#### 【0010】

##### 【発明が解決しようとする課題】

検証の対象のひとつである暗号通信手順を形式言語で表現する場合には、正則性の条件を満すことが必要であった。このために、従来までの形式言語理論によるアプローチでは、正則性の条件を満たさない暗号通信手順を、自動的に検証することはできなかった。

#### 【0011】

上記した3つの論文に提案されているいずれの検証方法も、正則性の条件を満たさない暗号通信手順については、秘密保持性を近似的に検証することは可能であっても、厳密な検証を行うことはできないという問題がある。

#### 【0012】

このことは、暗号通信手順に限らず、一般的なリアクティブ・システムの動作手順に関しても生じる問題である。

#### 【0013】

本発明は、正則性の条件を満たすか否かによらず、等式付ツリー・オートマトン理論の範疇にあるリアクティブ・システムの動作手順に関して、近似ではなく、厳密に安全性を検証することができるリアクティブ・システムの安全性検証装置、検証方法、検証プログラム及びそのプログラムを記録したコンピュータ読取可能な記録媒体を提供することを目的とする。

#### 【0014】

##### 【課題を解決するための手段】

本発明の目的は、以下の手段によって達成される。

**【0015】**

即ち、本発明の第1の態様によれば、関数記号の集合、書換規則の集合、公理の集合、項の集合、及び検査対象となる項の集合によって表わされるリアクティブ・システムの安全性検証装置であって、前記公理の集合が、交換則及び結合則のみを要素とする集合であり、前記公理の集合の下で、前記項の集合を受理する第1の等式付ツリー・オートマトンを生成する翻訳部、前記第1の等式付ツリー・オートマトンを初期データとして、前記書換規則の集合及び前記公理の集合の下で、前記項の集合から派生する項からなる集合と前記項の集合とを受理する第2の等式付ツリー・オートマトンを生成するシミュレーション部、及び前記第2の等式付ツリー・オートマトンと、前記検査対象となる項の集合を受理する第3の等式付ツリー・オートマトンとを結合して第4の等式付ツリー・オートマトンを生成し、該第4の等式付ツリー・オートマトンが受理する集合が空集合か否かを判断する集合演算部を備えていることを特徴とするリアクティブ・システムの安全性検証装置を提供することができる。

**【0016】**

本発明の第2の態様によれば、関数記号の集合、書換規則の集合、公理の集合、項の集合、及び検査対象となる項の集合によって表わされるリアクティブ・システムの安全性検証装置であって、前記公理の集合が、交換則及び結合則のみを要素とする集合であり、前記公理の集合の下で、前記項の集合を受理する第1の等式付ツリー・オートマトンを生成する翻訳部、前記第1の等式付ツリー・オートマトンを初期データとして、前記書換規則の集合及び前記公理の集合の下で、前記項の集合から派生する項からなる集合と前記項の集合とを受理する第2の等式付ツリー・オートマトンを生成するシミュレーション部、及び前記第2の等式付ツリー・オートマトンが、前記検査対象となる項を受理するか否かを判断する集合演算部を備えていることを特徴とするリアクティブ・システムの安全性検証装置を提供することができる。

**【0017】**

本発明の第3の態様によれば、関数記号の集合、書換規則の集合、公理の集合、項の集合、及び検査対象となる項の集合によって表わされるリアクティブ・シ

ステムの安全性検証方法であって、前記公理の集合が、交換則及び結合則のみを要素とする集合であり、前記公理の集合の下で、前記項の集合を受理する第1の等式付ツリー・オートマトンを生成する第1のステップ、前記第1の等式付ツリー・オートマトンを初期データとして、前記書換規則の集合及び前記公理の集合の下で、前記項の集合から派生する項からなる集合と前記項の集合とを受理する第2の等式付ツリー・オートマトンを生成する第2のステップ、及び前記第2の等式付ツリー・オートマトンと、前記検査対象となる項の集合を受理する第3の等式付ツリー・オートマトンとを結合して第4の等式付ツリー・オートマトンを生成し、該第4の等式付ツリー・オートマトンが受理する集合が空集合か否かを判断する第3のステップを含むことを特徴とするリアクティブ・システムの安全性検証方法を提供することができる。

#### 【0018】

本発明の第4の態様によれば、関数記号の集合、書換規則の集合、公理の集合、項の集合、及び検査対象となる項の集合によって表わされるリアクティブ・システムの安全性検証方法であって、前記公理の集合が、交換則及び結合則のみを要素とする集合であり、前記公理の集合の下で、前記項の集合を受理する第1の等式付ツリー・オートマトンを生成する第1のステップ、前記第1の等式付ツリー・オートマトンを初期データとして、前記書換規則の集合及び前記公理の集合の下で、前記項の集合から派生する項からなる集合と前記項の集合とを受理する第2の等式付ツリー・オートマトンを生成する第2のステップ、及び前記第2の等式付ツリー・オートマトンが、前記検査対象となる項を受理するか否かを判断する第3のステップを含むことを特徴とするリアクティブ・システムの安全性検証方法を提供することができる。

#### 【0019】

本発明の第5の態様によれば、コンピュータに、関数記号の集合、書換規則の集合、公理の集合、項の集合、及び検査対象となる項の集合によって表わされた手順の入力を受け付ける機能、交換則及び結合則のみを要素とする前記公理の集合の下で、前記項の集合を受理する第1の等式付ツリー・オートマトンを生成する機能、前記第1の等式付ツリー・オートマトンを初期データとして、前記書換

規則の集合及び前記公理の集合の下で、前記項の集合から派生する項からなる集合と前記項の集合とを受理する第2の等式付ツリー・オートマトンを生成する機能、及び前記第2の等式付ツリー・オートマトンと、前記検査対象となる項の集合を受理する第3の等式付ツリー・オートマトンとを結合して第4の等式付ツリー・オートマトンを生成し、該第4の等式付ツリー・オートマトンが受理する集合が空集合か否かを判断する機能を実現させるためのリアクティブ・システムの安全性検証プログラムを提供することができる。

#### 【0020】

本発明の第6の態様によれば、コンピュータに、関数記号の集合、書換規則の集合、公理の集合、項の集合、及び検査対象となる項の集合によって表わされた手順の入力を受け付ける機能、交換則及び結合則のみを要素とする前記公理の集合の下で、前記項の集合を受理する第1の等式付ツリー・オートマトンを生成する機能、前記第1の等式付ツリー・オートマトンを初期データとして、前記書換規則の集合及び前記公理の集合の下で、前記項の集合から派生する項からなる集合と前記項の集合とを受理する第2の等式付ツリー・オートマトンを生成する機能、及び前記第2の等式付ツリー・オートマトンが、前記検査対象となる項を受理するか否かを判断する機能を実現させるためのリアクティブ・システムの安全性検証プログラムを提供することができる。

#### 【0021】

前記第1～第6の態様において、前記関数記号の集合が、暗号処理、復号処理、及び通信処理を表す関数を要素として含む集合であり、前記書換規則の集合が、暗号処理された情報が復号処理されることによって平文に戻ることを表す規則を要素として含む集合であり、前記検査対象となる項が、秘密情報であり、前記項の集合が、秘密情報を交換する複数の主体の各々の知識の集合、及びこれら複数の主体間で交換される情報を傍受する主体の知識の集合であることができる。

#### 【0022】

本発明の第7の態様によれば、上記した何れかのプログラムを記録したコンピュータ読取可能な記録媒体を提供することができる。

#### 【0023】

**【発明の実施の形態】**

以下、本発明に係る実施の形態に関して、添付図を参照して説明する。図1は、本発明の実施の形態に係るリアクティブ・システムの安全性検証装置の構成を示すブロック図である。本実施の形態に係るリアクティブ・システムの安全性検証装置は、各構成部を制御する中央演算処理部（以下、CPUと記す）1、データを一時的に記録する一時記憶部（以下、メモリと記す）2、データを持続的に記録する記録部3、各構成部の間でデータを交換するための内部バス4、データの入力を受け付ける入力部5、処理結果などを表示する表示部6、インターフェース部7、翻訳部8、シミュレーション部9、及び集合演算部10を備えている。

**【0024】**

CPU1は、各構成部に対する制御を行い、翻訳部8、シミュレーション部9、及び集合演算部10は、CPU1からの命令を受けて、後述するように安全性検証における中心的処理を行う。検証の対象である動作手順に関する情報は、入力部5から入力され、インターフェース部7によって所定のデータ形式に変換され、記録部3に記録される。後述するデータ処理において、記録部3から必要なデータがメモリ2に読み出され、メモリ2上で処理が実行される。処理結果は記録部3に記録され、必要に応じて表示部6に表示される。

**【0025】**

図2は、暗号通信手順に関して、本実施の形態に係るリアクティブ・システムの安全性検証装置の処理を示すフローチャートである。図1及び2に基づき、検証装置の処理内容を説明する前に、検証の対象のひとつである暗号通信手順の記述方法に関して説明する。

**【0026】**

有限の要素からなる集合を対象とした処理は、直接各要素を記述し、これに対する処理を行うことが可能であるが、本発明が検証の対象とする暗号通信手順は、一般に有限の要素からなる集合として記述することはできない。従って、検証の対象を、取り得る状態の集合と、状態間の遷移を引き起こす入力とによって記述された「系」、即ちオートマトンを用いて表す。

**【0027】**

具体的には、検証の対象となる暗号通信手順を次の式1に示す5個の記号の組で表現する。このことは、式1の記号表現によって表現することが可能な「通信手順」のみが検証の対象となり得ることを意味するものもある。以下において、記号「|」は集合を表わし、例えば  $\{L_i\}$  は  $L_1, L_2, L_3$ 、などを構成要素とする集合を表すこととする。

$$P = (F, \{R_i\}, U, \{K_i\}, S[j \rightarrow k]) \dots \quad (\text{式1})$$

ここで、Pは暗号通信手順を表わし、iは、mを自然数として、 $1 \leq i \leq m$ を満たし、j、kはm以下の任意の正の整数とする。

#### 【0028】

F、R<sub>i</sub>、U、K<sub>i</sub>、S[j → k]は、全て集合を表す。Fは暗号処理、復号処理、通信処理（鍵交換、メッセージ交換など）、通信データ（メッセージ、パスワードなど）などを表す「関数記号」の集合、R<sub>i</sub>（i = 1 ~ m）は「書換規則」の集合、Uは「公理」の集合、K<sub>i</sub>（i = 1 ~ m）は「項」の集合（例えば、情報の交換に関係する主体がA、B、Cの3者であれば、各々に関してK<sub>1</sub>、K<sub>2</sub>、K<sub>3</sub>が設定される）を表わしている。ここで、「項」とは、集合の構成要素を意味し、単一の記号または複数の記号の組合せで表現される。S[j → k]は、K<sub>j</sub>を所有する主体がK<sub>k</sub>を所有する主体に対して秘密にしたいメッセージの集合（以下、検査対象となる項の集合と記す）を表している。

#### 【0029】

公理の集合Uは、交換則、結合則を要素とする集合であり、項x、yに対する演算子を「+」とすると、

$$\text{交換則: } x + y = y + x$$

$$\text{結合則: } (x + y) + z = x + (y + z)$$

と表すことができる。ここで「+」は加算を表す記号ではない。

#### 【0030】

集合K<sub>i</sub>は、暗号情報を交換する複数の主体及び通信の傍受者の各々が獲得できる知識の集合を表わしている（以下、ナレッジと記す）。例えば、主体A、Bが暗号情報を相互に交換する場合、主体A、Bのナレッジを各々K<sub>1</sub>、K<sub>2</sub>とし、傍受者CのナレッジをK<sub>3</sub>と表す。ナレッジは初期状態では有限個数の項の集

合であるが、一般に時間経過によって拡張される。例えば傍受者CのナレッジK<sub>3</sub>は、傍受者Cが主体A、Bの間で繰り返し交換される暗号情報を傍受することに伴って、増加する。

#### 【0031】

書換規則 {R<sub>i</sub>} は、項に対して関数記号を適用した記述が、どのような異なる記述に書き換えられるかを指定した規則である。例えば、暗号処理をE、復号処理をD、鍵をx、平文メッセージをyとすると、「暗号化メッセージをその暗号鍵で復号すると元の平文メッセージが得られる」という性質は、書換規則「D(x, E(x, y)) → y」によって表わされる。ここで矢印は、矢印の左側の項を右側の項で書き換えることができることを表わしている。

#### 【0032】

以下において、図2に基づき、暗号通信手順に関して、本実施の形態に係るリアクティブ・システムの安全性検証装置が行う処理を説明する。

#### 【0033】

まず、ステップ20において、CPU1は、入力部5を介して、検証対象となる手順に関する情報、即ち上記した記号による表記(F, {R<sub>i</sub>}, U, {K<sub>i</sub>}, S[j → k])の入力を受け付け、取得した入力データを記録部3に記録する。このとき、検査対象となる項の集合S[j → k]が無限集合の場合には、その集合を受理するオートマトンの記述が入力される。取得したデータは、論理記号を使用した論理式であり、例えばテキストデータ形式で入力され、以降の処理においてはテキスト形式のまま処理される。

#### 【0034】

ステップ21において、CPU1は、ステップ20で記録部3に記録したデータを記録部3からメモリ2上の所定領域に読み出し、翻訳開始の命令コード及び必要なデータのメモリアドレス情報を翻訳部8に伝送する。

#### 【0035】

ステップ22において、命令コードを受信した翻訳部8は、メモリアドレス情報に基づいてメモリ2からナレッジ{K<sub>i</sub>}及び公理Uを取得し、各々のナレッジK<sub>i</sub>及び公理Uを使用して等式付ツリー・オートマトンA<sub>i</sub>を生成する。生成

された等式付ツリー・オートマトン  $A_i$  は、CPU1 を介してメモリ2 の所定領域に記録される。

#### 【0036】

ナレッジ  $K_i$  の項を受理するオートマトン  $A_i$  の生成方法は、楫勇一、藤原融、嵩忠雄による “Solving a Unification Problem under Constrained Substitution using Tree Automata(ツリー・オートマトンを用いた制約付代入における单一化問題の解法)” (Journal of Symbolic Computation 23(1), pp.79-117, 1997) などによって公知であり、この分野の通常の知識を有する者にとって容易であるので、ここでは記載を省略する。

#### 【0037】

翻訳部8が output する各々の  $A_i$  は、公理 U の下で、対応する集合であるナレッジ  $K_i$  を受理する等式付ツリー・オートマトンである。ここで、上記したように、オートマトンとは、取り得る状態と入力による状態間の遷移とによって、システムを表現した系であり、ツリー・オートマトンとは、ツリー構造を有するデータ（項）を受理するオートマトンである。

#### 【0038】

また、「等式付」とは、ツリー構造を有するデータの間に、等価性の概念を表現する公理が成立することを表している。例えば、項「 $1 + 2$ 」と項「 $2 + 1$ 」とを等価とみなすためには、公理「 $x + y = y + x$ 」が必要となる。同様に、項「 $(1 + 2) + 3$ 」と項「 $3 + (2 + 1)$ 」とを等価とみなすためには、 $x + y = y + x$  を仮定すれば十分である。公理「 $(x + y) + z = x + (y + z)$ 」が必要となるのは、例えば項「 $(1 + 2) + 3$ 」と項「 $1 + (2 + 3)$ 」とを等価とみなす場合である。ここで、「+」は演算子であり、加算を意味するものではない。即ち、「等式付ツリー・オートマトン」とは、ツリー構造を有するデータの間に等価性の公理が成り立つことを前提として、ツリー構造のデータの受理／不受理を決定するオートマトンである。

#### 【0039】

また、「集合を受理する」とは、その集合の要素のみを受理する、即ち、その集合の要素（項）を全て受理し、それ以外を受理しないことを意味する。

**【0040】**

以上のことから、各々のオートマトン  $A_i$  は、各項がツリー構造を有し、項の間に等価性の公理  $U$  が成り立つナレッジ  $K_i$  を受理する等式付ツリー・オートマトンとして記述される。

**【0041】**

以上のように、ステップ22において、翻訳部8は、各々の主体の初期のナレッジ  $K_i$  を等式付ツリー・オートマトン  $A_i$  に変換、即ち、集合  $K_i$  を受理する等式付ツリー・オートマトン  $A_i$  として記述する。これによって、集合  $K_i$  に関する処理を、その集合と等価な等式付ツリー・オートマトン  $A_i$  に関する処理として扱うことができる。

**【0042】**

ステップ23において、CPU1は、シミュレーション開始の命令コードを、翻訳部8によって生成された等式付ツリー・オートマトン  $\{A_i\}$  、ナレッジ  $\{K_i\}$  及び公理  $U$  が記録されたメモリ2上のアドレス情報と共に、シミュレーション部9に伝送する。

**【0043】**

ステップ24において、CPU1からの開始命令を受信したシミュレーション部9は、受信したメモリアドレス情報に基づいてメモリ2から等式付ツリー・オートマトン  $\{A_i\}$  、ナレッジ  $\{K_i\}$  、公理  $U$  を取得し、所定の処理を実行する。

**【0044】**

即ち、シミュレーション部9は、メモリ2から読み出した等式付ツリー・オートマトン  $\{A_i\}$  を初期データとして、繰り返し処理によって、拡張されるナレッジを受理する等式付ツリー・オートマトンを生成する。シミュレーション部9は、処理の経過途中において所定の収束条件が満たされたと判断した場合、例えば繰り返し処理の結果、等式付ツリー・オートマトンが変化しなくなったと判断した場合、その時点の等式付ツリー・オートマトン  $\{A_i*\}$  を出力し、収束したことを知らせる収束コードと共にCPU1に伝送する。CPU1は、シミュレーション部9から受信したこれらの算出データを、メモリ2の所定領域に記録す

る。

#### 【0045】

図3は、上記した等式付ツリー・オートマトン  $\{A_i*\}$  の生成において、繰り返し処理で呼び出されるサブルーチンのアルゴリズムを示した図である。このアルゴリズムは、入出力引数の指定、初期設定、第1処理、第2処理から構成されている。入力引数は、書換規則の1つの要素 ( $1 \rightarrow r$ )、及び等式付ツリー・オートマトン (A/AC) であり、出力引数は計算結果の等式付ツリー・オートマトン ( $B_{1 \rightarrow r}/AC$ ) である。

#### 【0046】

初期設定では、以降の処理に対する初期値として、A0に入力引数の値であるツリー・オートマトンAをセットし、集合S、Tをセットする。集合S、Tは、入力引数の値である書換規則  $1 \rightarrow r$  の左辺 (1に相当する)、右辺 (rに相当する) の項を、ツリー構造として記述した場合の位置情報を要素とする。

#### 【0047】

第1処理では、集合Sの要素に基づいて所定の順序で遷移規則を追加、変更する。第2処理では、集合Tの要素に基づいて所定の順序で処理して遷移規則を追加、変更する。第1処理及び第2処理は、同一の原理に基づいて計算を行うので、第1処理の場合についてのみ説明する。

#### 【0048】

第1処理において、まず、所定条件を満たす要素pを集合Sから選択する。この条件は、要素pがツリー構造の末端に位置する条件である。

#### 【0049】

次に、着目した要素pの関数記号をf、その引数部分の項を  $t_1, \dots, t_n$  として、要素pに相当する部分の項が  $f(t_1, \dots, t_n)$  と表せるとき、次の書換規則に従って、 $L(A_{i+1}/AC)$  に含まれるすべての項に対して書換を行うことによって、項の集合  $L(A_{i+1}/AC)$  を得る。

書換規則：  $f(c^{p,1}t_1, \dots, c^{p,n}t_n) \rightarrow c^{p,1}|p$

項の集合  $L(A_{i+1}/AC)$  は、fが公理に用いられている関数記号である場合には、Rohit Parikhの手法をツリー構造のデータ用に拡張した方法で計算すること

ができる。 $f$  が公理に用いられている関数記号ではない場合には、上記した楫勇一らの論文に紹介されている方法で、 $A_i$  を基に  $A_{i+1}$  を計算することができる。以降繰り返し条件を満たすまで計算すれば、書換規則の左辺（1に相当する）に合致する項の前処理が終了し、次の処理に移る。

#### 【0050】

第2処理に関しても、所定条件を満たす要素  $q$  を集合  $T$  から選択し、同様に処理することによって、第2処理のループ計算が繰り返し条件を満たして、等式付ツリー・オートマトン  $B_j/AC$  が得られるので、これを  $B_{1 \rightarrow r}/AC$  として第2処理を終了する。以上で  $B_{1 \rightarrow r}/AC$  が得られ、これが出力引数の値となる。

#### 【0051】

以上の様に、ここでの処理には、Rohit Parikhによる文字列空間からベクトル空間への写像を用いた手法を、ツリー構造のデータを扱えるように開発した方法を用いている。Parikhの提案した方法は、"On Context-Free Languages" (Journal of the ACM 13(4), pp.570-581, 1966) に開示されているので、ここでは記載を省略する。

#### 【0052】

書換規則の複数の要素の各々に対して、等式付ツリー・オートマトン  $A_i$  を入力引数  $A/AC$  として、上記した処理を1回行い、それらの結果を集めることによって、1回の処理に相当して拡張されたナレッジを受理する等式付ツリー・オートマトン  $A_i * (1)$  が得られる。得られた等式付ツリー・オートマトン  $A_i * (1)$  を入力引数  $A/AC$  として、再度同様に処理することによって、2回の処理に相当して拡張されたナレッジを受理する等式付ツリー・オートマトン  $A_i * (2)$  が得られる。この処理を繰り返すことによって、次々と拡張されたナレッジを受理する等式付ツリー・オートマトン  $A_i * (n)$  が得られる。

#### 【0053】

シミュレータ部は、上記したように、等式付ツリー・オートマトン  $A_i * (n)$  が、1回前の等式付ツリー・オートマトン  $A_i * (n-1)$  から変化したか否かによって、収束を判断する。収束したと判断した場合、 $A_i * (n)$  を、公理  $U$  及び書換規則  $R_i$  の条件の下で、初期のナレッジ  $K_i$  から派生する全ての項か

らなる集合と初期のナレッジ  $K_i$  を受理する等式付ツリー・オートマトン  $A_i^*$  とする。

#### 【0054】

主体が暗号情報を交換することによって、各々の主体及び傍受者のナレッジ  $\{K_i\}$  は徐々に増大して行くが、シミュレーション部9はステップ24において、所定の条件(公理U及び書換規則  $\{R_i\}$ )の下で、この到達可能な最大のナレッジ  $\{K_i\}$  を等式付ツリー・オートマトン  $\{A_i^*\}$  として記述する。これは、無限集合の境界を確定することに相当し、等式付ツリー・オートマトンとして記述することによって可能となる。

#### 【0055】

シミュレーション部9は、所定の時間若しくは所定回数の繰り返し処理を経過しても収束と判断できなかった場合、収束しなかったことを知らせる非収束コードをCPU1に伝送し、処理を終了する。これは、その暗号通信手順の検証ができないことを意味する。

#### 【0056】

ステップ25において、CPU1は、シミュレーション部9から受信したコードが、収束コードか否かを判断し、収束コードと判断した場合にステップ26に移行し、非収束コードと判断した場合にはステップ31に移行する。

#### 【0057】

ステップ26において、CPU1は、検査対象となる項の集合  $S[j \rightarrow k]$  が有限集合か否かを判断し、有限集合と判断した場合、ステップ27に移行し、有限集合でないと判断した場合、ステップ28に移行する。ステップ20において説明したように、検査対象となる項の集合  $S[j \rightarrow k]$  が有限集合でない場合には、検査対象となる項の集合  $S[j \rightarrow k]$  を受理する等式付ツリー・オートマトンの記述が入力されている。

#### 【0058】

秘密情報が有限である場合、ステップ27において、演算部は、傍受者Cの最大ナレッジを表す等式付ツリー・オートマトン  $A_3^*$ 、及び検査対象となる項の集合  $S[j \rightarrow k]$ 、例えば  $S[1 \rightarrow 3]$  をメモリ2から読み出し、各々の要素が

等式付ツリー・オートマトンA 3 \*に受理されるか否かを判断し、その結果に応じたデータをメモリ2の所定領域に記録する。このとき、検査対象となる項の集合S [1→3] のいずれの要素も等式付ツリー・オートマトンA 3 \*によって受理されなかった場合、記録されるデータは“0”であり、要素の中の少なくとも1つが等式付ツリー・オートマトンA 3 \*によって受理された場合、記録されるデータは“1”である。

#### 【0059】

検査対象となる項の集合S [j→k] が有限集合でない場合、ステップ28において、CPU1は集合演算部10に開始命令及び必要なメモリアドレス情報を送信する。

#### 【0060】

ステップ29において、集合演算部10は、CPU1から受信したメモリアドレス情報に基づいてメモリ2から、傍受者Cの可能な最大ナレッジを表す等式付ツリー・オートマトンA 3 \*、及び検査対象となる項の集合を受理する等式付ツリー・オートマトンS [j→k]、例えば主体Aから傍受者Cに対する検査対象となる項の集合を受理する等式付ツリー・オートマトンS [1→3] を取得し、A 3 \*とS [1→3] とを合成して等式付ツリー・オートマトンWを生成する。この合成は、各々のオートマトンによって受理される2つの集合(A 3 \*とS [1→3])の積(A 3 \*∩S [1→3])を求めるに相当する。即ち、2つの集合の共通部分(A 3 \*∩S [1→3])は、等式付ツリー・オートマトンWに受理される。

#### 【0061】

図4は、2つのツリー・オートマトンを合成する方法の一例を説明する図である。結合則及び交換則が成り立つ集合を受理する2つのオートマトン(A/AC、B/AC)を結合したオートマトンの遷移規則は、図4に示した4種類のRx～Rgを合わせたものとなる。詳細は、本願発明者による公知文献“Beyond Regularity : Equational Tree Automata for Associative and Commutative Theories(正則性を超えて：結合・交換理論に関する等式付ツリー・オートマトン)”(Proceedings of 15th International Conference of the European Association for

r Computer Science Logic, Paris (France), Lecture Notes in Computer Science 2142, pp. 539-553, 2001.) において開示されているので、ここでは記載を省略する。

#### 【0062】

ステップ30において、集合演算部10は、ステップ29で得られた等式付ツリー・オートマトンWが受理する集合 $A3 * \cap S [1 \rightarrow 3]$ が空（くう：構成要素が存在しないこと）であるか否かを判断する。集合演算部10は、空集合と判断すれば空コードをCPU1に伝送し、空集合でないと判断すれば非空コードをCPU1に伝送する。

#### 【0063】

空の判定は、項全体を受理する等式付ツリー・オートマトンB/ACの終了状態 $q$ から、判定対象である等式付ツリー・オートマトンA/ACの終了状態 $p$ に到達可能か否かを判断することによって行われる。具体的には、ツリー・オートマトンBの遷移規則の左辺と右辺を入れ替え、これによって得られるツリー・オートマトンを $B^{-1}$ とする。 $(A \cup B^{-1}) / AC$ を基底AC書換系とみなしたとき、状態 $q$ から状態 $p$ への到達可能性は、等式付ツリー・オートマトンA/ACが何らかの要素を受理することと等価になる。

#### 【0064】

終了状態 $q$ から終了状態 $p$ に到達する経路を計算する具体的方法は、Richard MayerとMichael Rusinowitchによる“Reachability is Decidable for Ground AC Rewrite Systems (基底AC書換システムの到達可能性)” (Proceedings of 3rd International Workshop on Verification of Infinite State Systems, Aalborg (Denmark), 1998) に開示されているので、ここでは記載を省略する。

#### 【0065】

ステップ31において、CPU1は、表示部6に、以上のステップでの処理結果に応じた表示をする。即ち、CPU1は、非収束コード（ステップ25）を受信した場合、対象の暗号通信手順の検証ができないことを表示する。CPU1は、ステップS27での処理の結果としてメモリ2に記録されたデータが“0”（非受理を表す）の場合、検証対象の暗号通信手順が、その受理されなかった秘密

情報を通信する限りにおいて安全であることを表示し、“1”（受理を表す）の場合、検証対象の暗号通信手順が安全ではないことを表示する。CPU1は、ステップ30での処理の結果、空コードを受信した場合、検証対象の暗号通信手順が安全であることを表示し、非空コードを受信した場合、検証対象の暗号通信手順は安全ではないことを表示する。以上の処理の後、CPU1は、必要に応じて、メモリ2上に一時記録されたデータを記録部3に記録し、暗号通信手順の安全性検証を終了する。

#### 【0066】

以下において、具体的な暗号通信手順への本発明の適用に関して説明する。

##### （実施例1）

本発明をDiffie-Hellman型暗号通信手順に適用する場合について説明する。Diffie-Hellman型暗号通信手順とは、主体A、Bの間で暗号通信する場合に、暗号・復号用の鍵として秘密鍵と公開鍵の2種類の鍵を所定の規則に従って生成し、公開鍵を主体A、Bの間で、例えば通信ネットワークを介して交換する暗号通信手順である。

#### 【0067】

Diffie-Hellman型暗号通信手順において、主体Aは、任意に選択した大きい正の整数xを基に、次の式2によって公開鍵Xを生成し、Xを主体Bに通信する。主体Bは、任意に選択した大きい正の整数yを基に、次の式3によって公開鍵Yを生成し、Yを主体Aに通信する。

$$X = g^x \bmod n \quad \dots \quad (\text{式2})$$

$$Y = g^y \bmod n \quad \dots \quad (\text{式3})$$

ここで、g、nは任意の大きい素数であり、「 $\bmod$ 」は剰余を表す。例えば、 $a \bmod b$ は、aをbで除した場合の剰余を表わしている。

#### 【0068】

主体Aは、主体Bから取得したY及び自己が選択したxを使用して次の式4によってkを計算することができ、また、主体Bは、主体Aから取得したX及び自己が選択したyを使用して次の式5によってk'を計算することができる。

$$k = Y^x \bmod n \quad \dots \quad (\text{式4})$$

$$k' = XY \bmod n \quad \dots \quad (\text{式5})$$

計算結果の  $k$  及び  $k'$  は、何れも  $g^{xy} \bmod n$  に等しいことから、主体A、Bは共通の数値を得ることができ、この値は、 $x$  または  $y$  が分からなければ、主体A、B間で交換される  $g$ 、 $n$ 、 $X$ 、 $Y$  から取得することが非常に困難である。従って、主体A、Bは、 $k$  ( $= k'$ ) を、暗号・復号用の鍵として使用することによって、安全に秘密データの交換が可能となる。

#### 【0069】

Diffie-Hellman型暗号通信手順は、上記した手順以外にも様々なバリエーションがあるが、それらは全て以下のように表現することができる。主体AとBとでメッセージMをDiffie-Hellman型暗号通信手順によって、暗号化して交換し、傍受者をCで表わし、Diffie-Hellman型暗号通信手順に共通する「振る舞い」を「+」で表現する。

#### 【0070】

暗号通信手順Pは、

$$P = (F, \{R_i\}, U, \{K_i\}, S [1 \rightarrow 3])$$

と表現できる。ここで、 $i = 1 \sim 3$  であり、各記号は、式1と同じ意味である。

#### 【0071】

関数記号の集合Fは、

$$F = \{A(0), B(0), C(0), N(0), M(0), k(1), + (2), E(2), D(2)\}$$

となる。ここで、A、Bは暗号情報を交換する主体、Cは傍受者であり、Nは任意の自然数、Mは暗号化された後に交換される秘密情報（例えば、初期に主体Aのみが知っている情報で、暗号化されて主体Bに通信される情報）であり、kは鍵、Eは暗号処理、Dは復号処理を表す。「+」は加算を表す記号ではなく、上記したようにDiffie-Hellman型暗号通信手順において共通する「振る舞い」を表す記号である。カッコ内の数字は各記号の引数、即ち決定に必要な「変数」の数である。従って、A(0)、B(0)、C(0)、N(0)、M(0)は所定の記号又は数値であり、これらを決定するために変数は必要ない。k(1)は、決定に1つの変数が必要であり、E、D、+には各々2つの変数が必要である。例

えば、 $k(1)$  は、主体A、Bのいずれかを決めれば、 $k(A)$  又は $k(B)$  として決定される。Eは、暗号処理の対象となる項と鍵とによって決定される。Dは、復号処理の対象となる項と鍵とによって決定される。+は、mod関数の引数、即ち除数と被除数とによって決定される。

#### 【0072】

書換規則  $\{R_i\}$  は、

$$R1 = R2 = R3 = \{D(x, E(x, y)) \rightarrow y\}$$

である。これは、yを暗号処理した結果であるE(x, y)を復号処理すれば、yを得ることができることを表す。

#### 【0073】

公理Uは、

$$U = \{x + y = y + x, (x + y) + z = x + (y + z)\}$$

である。ここでも、「+」は加算ではなく、Diffie-Hellman型暗号通信手順において共通する「振る舞い」を表す。即ち、Diffie-Hellman型暗号通信手順において共通する「振る舞い」は、交換則、結合則を満たすことを表わしている。

#### 【0074】

ナレッジ  $\{K_i\}$  は、

$$K1 = \{A, B, k(A) + k(B) + N, k(A), N, M\}$$

$$K2 = \{A, B, k(A) + k(B) + N, k(B), N, E(k(A) + k(B) + N, M)\}$$

$$K3 = \{A, B, C, k(A) + N, k(B) + N, k(C), N, E(k(A) + k(B) + N, M)\}$$

である。

#### 【0075】

$K1 \sim K3$  は、それぞれ主体A～Cのナレッジである。主体AとBとの間で、鍵の情報として $N$ 、 $k(A) + N$ 、 $k(B) + N$ 、及び、暗号化されたメッセージ $E(k(A) + k(B) + N, M)$  が交換されることから、傍受者Cのナレッジ $K3$ にはそれらの情報が含まれている。傍受者Cのナレッジ $K3$ において、 $k(A) + N$ 、 $k(B) + N$  と表記されているが、主体AとBとの間の暗号通

信において、 $k(A) + N$ 、 $k(B) + N$  はそれぞれ1つの情報として交換される。従って、傍受者Cは $k(A) + N$ を1つの情報として取得することは可能であるが、その構成、即ち $k(A)$ と $N$ とから生成されていることを直接知ることはできない。 $E(k(A) + k(B) + N, M)$  に関しても同様に、傍受者Cは $E(k(A) + k(B) + N, M)$ を1つの情報として取得できるだけで、その構成を直接知ることはできない。

#### 【0076】

主体Aから傍受者Cに対する検査対象となる項の集合 $S[1 \rightarrow 3]$ は、

$$S[1 \rightarrow 3] = \{M\}$$

である。

#### 【0077】

上記した代表的なDiffie-Hellman型暗号通信手順との対応関係は次の様になる

。

- ・ 主体Aが任意に選択した整数 $x$ は、 $k(A)$ に対応
- ・ 主体Bが任意に選択した整数 $y$ は、 $k(B)$ に対応
- ・  $X (X = g^x \bmod n)$  は、 $k(A) + N$ に対応
- ・  $Y (Y = g^y \bmod n)$  は、 $k(B) + N$ に対応
- ・  $k (k = Y^x \bmod n)$  は、 $k(A) + (k(B) + N)$ に対応
- ・  $k' (k' = X^y \bmod n)$  は、 $k(B) + (k(A) + N)$ に対応

また、 $k = k'$  の等価性から、 $k(A) + (k(B) + N) = k(B) + (k(A) + N)$  が導かれる。ここで、 $a + N$ は、 $g^a \bmod n$  を表すと仮定している。

#### 【0078】

以上で記述した、暗号通信手順P、関数記号の集合F、書換規則 $\{R_i\}$ 、ナレッジ $\{K_i\}$ 、公理U、及び検査対象となる項の集合 $S[1 \rightarrow 3]$ を、入力部5から入力することによって、図2に示したように、等式付ツリー・オートマトンが生成され、これを用いて計算が自動的に実行され、暗号通信手順の安全性が判断される。本実施例の場合には、上記したように、秘密情報は引数のない関数記号M（定数関数記号）として集合Fに含まれている。即ち、Mは項である。また

、検査対象はMだけからなる有限集合であるから、ステップ26における判定の結果、ステップ27の処理が実行される。即ち、暗号通信手順の安全性は、「A<sub>3\*</sub>（傍受者Cが最終的に持ち得る知識の集合を受理する等式付ツリー・オートマトン）の中に、M（秘密情報）が含まれるか否か」という判断により行なわれる。

#### 【0079】

##### （実施例2）

本発明を、Shamirによって提案されたワンタイム・パッド（One-Time Pad）を用いた暗号通信手順に適用する場合について説明する。

#### 【0080】

ワンタイム・パッドの暗号通信手順は、交換則  $E(k(A), E(k(B), M)) = E(k(B), E(k(A), M))$  を満たす暗号処理の下で、次の手順（1）～（4）で行われる。

- (1) 主体Aは、鍵  $k(A)$  を使用して秘密情報Mを暗号処理して得られた  $E(k(A), M)$  を、主体Bに送信する。
- (2) 主体Bは、取得した  $E(k(A), M)$  を鍵  $k(B)$  を使用して暗号処理して得られた  $E(k(B), E(k(A), M))$  を、主体Aに送信する。
- (3) 主体Aは、取得した  $E(k(B), E(k(A), M))$  を鍵  $k(A)$  を使用して復号処理して得られた  $D(k(A), E(k(B), E(k(A), M)))$  を、主体Bに送信する。
- (4) 主体Bは、取得した  $D(k(A), E(k(B), E(k(A), M)))$  を鍵  $k(B)$  を使用して復号処理することによって秘密情報Mを取得することができる。即ち、交換則を考慮すれば、 $D(k(B), D(k(A), E(k(B), E(k(A), M)))) = D(k(B), D(k(A), E(k(A), E(k(B), M)))) = D(k(B), E(k(B), M)) = M$  となる。

#### 【0081】

交換則を満たす暗号／復号処理として、暗号化しようとする平文に対して同じ長さの乱数ビット列を鍵として、これと平文との排他的論理和演算（以下、XO

Rと記す) が使用される。

#### 【0082】

この場合、暗号通信手順P、関数記号の集合F、公理U、及び検査対象となる項の集合S[1→3]は、「+」がXORを表すとすれば、上記した実施例1の場合と同じ表記となる。

#### 【0083】

書換規則 {R i} は、

$$E(x, E(y, z)) \rightarrow E(x+y, z)$$

$$E(id, x) \rightarrow x$$

$$id + id \rightarrow id$$

$$(x+x) + y \rightarrow y$$

と表わせる。idは、便宜上導入した関数である。

#### 【0084】

ナレッジ {K i} は、

$$K1 = \{A, B, k(A), id, M, E(k(B), E(k(A), M))\}$$

$$K2 = \{A, B, k(B), id, E(k(A), M), D(k(A), E(k(B), E(k(A), M)))\}$$

$$K3 = \{A, B, C, k(C), id, E(k(A), M), E(k(B), E(k(A), M)), D(k(A), E(k(B), E(k(A), M)))\}$$

である。

#### 【0085】

以上で記述した、暗号通信手順P、関数記号の集合F、書換規則 {R i} 、ナレッジ {K i} 、公理U、及び検査対象となる項の集合S [1→3] を、入力部5から入力することによって、図2に示したように、等式付ツリー・オートマトンが生成され、これを用いて計算が自動的に実行され、暗号通信手順の安全性が判断される。本実施例の場合にも、実施例1と同様に、秘密情報は引数のない関数記号Mとして集合Fに含まれていることから、ステップ27の処理が実行される。

#### 【0086】

上記した実施例1、2において、無限の秘密情報を対象とする場合には、予め求められた秘密情報の集合を受理する等式付ツリー・オートマトンSを入力することによって、ステップ28～30の処理によって安全性の検証が行われる。

#### 【0087】

以上において、検査対象となる項の集合S [ $j \rightarrow k$ ] が有限集合である場合には、ステップ27の処理が行われるが、この場合にもS [ $j \rightarrow k$ ] を等式付ツリー・オートマトンとして記述して、予めステップ20において入力しておくことによって、ステップ28～30の処理を行うようにすることができる。

#### 【0088】

また、以上において、検査対象となる項の集合が無限集合である場合には、ステップ20において等式付ツリー・オートマトンとして記述されたS [ $j \rightarrow k$ ] を入力することとしたが、この場合にも検査対象となる項の集合S [ $j \rightarrow k$ ] を入力し、翻訳部8によってこれを受理する等式付ツリー・オートマトンを生成するようにすることができる。

#### 【0089】

また、以上においては、厳密な安全性の検証ができるように、図2のステップ24の処理が収束しない場合、ステップ26～30の処理を行わないこととしたが、図5に示すように変更することも可能である。図5において、ステップ50～52、56～61の処理は、それぞれ図2のステップ20～22、26～31の処理と同じである。

#### 【0090】

ステップ53において、CPU1は、ステップ23と同様にシミュレーション開始の命令コード等をシミュレーション部9に伝送し、シミュレーション部9はカウンタに初期値、例えば“0”をセットする。

#### 【0091】

ステップ54において、シミュレーション部9は、ステップ24での処理に関して説明したように、拡張されるナレッジK<sub>i</sub>を受理する等式付ツリー・オートマトンA<sub>i</sub>\* (n)を生成し、ステップ53においてセットされたカウンタの初期値を1だけ変化、例えば1だけ増加させた後、ステップ55に移行する。

**【0092】**

ステップ55において、シミュレーション部9は、カウンタの値が予め指定した値（例えば自然数n0）であるか否かを判断する。指定値（n0）であればステップ56に移行し、指定値（n0）でなければステップ54に戻る。これによつて、指定回数（n0）だけステップ54の処理を行なわせることができ、対応して拡張されたナレッジKiを受理する等式付ツリー・オートマトンAi\*（n0）が生成される。

**【0093】**

図5においては、拡張されるナレッジKiを受理する等式付ツリー・オートマトンAi\*（n）の収束性を判断せずに、ステップ54の処理を所定回数行った後、常にステップ56以降の処理を行う。その結果、安全であると判断された場合には、その判断は必ずしも正しいものではなく、実際には安全でない可能性があるが、安全でないと判断された場合には、その判断は正しい。従つて、図5に示した処理も、暗号通信手順の安全性の検証において有効な処理である。

**【0094】**

以上において、暗号通信手順の安全性の検証に関して、実施の形態及び実施例を説明したが、その他のリアクティブ・システムに関しても同様に、安全性を検証することが可能である。例えば、原子炉や航空機などのシステムの制御手順に関する安全性を検証することが可能である。さらに、システムの規模によらず、設計の初期段階で安全性の検証をおこなうことができる。

**【0095】**

この場合には、翻訳部8による初期の知識を受理する等式付ツリー・オートマトンを生成する処理（図2のステップ22）は、対象のリアクティブ・システムを記述する等式付ツリー・オートマトンを生成する処理とすることができる。若しくは、予め生成された対象のリアクティブ・システムを記述する等式付ツリー・オートマトンをステップ20で入力する場合には、ステップ22における処理を省略することができる。

**【0096】**

その他の処理は、上記した実施の形態と同様であり、例えば、有限の検査対象

となる状態、即ち危険な状態の集合  $S [j \rightarrow k]$  が、ステップ24において生成された等式付ツリー・オートマトンによって受理されるか否かを判断することによって、リアクティブ・システムの安全性の検証が可能となる。検査対象となる状態の集合  $S [j \rightarrow k]$  が無限集合の場合には、等式付ツリーオートマトンとして記述された  $S [j \rightarrow k]$  を用いて、上記した実施の形態と同様に空の判定によって、リアクティブ・システムの安全性の検証が可能となる。

#### 【0097】

##### 【発明の効果】

本発明に係るリアクティブ・システムの安全性検証装置によって、暗号通信手順に関して、通信の傍受者が取得可能な最大の知識の集合に対応する等式付ツリー・オートマトンを生成することができ、秘密情報が傍受者の知識の集合の中に含まれ得るか否か、即ち暗号通信手順の安全性を検証することが可能となる。特に、正則性の条件を満たさない暗号通信手順に関しても、近似ではなく、厳密に暗号通信手順の安全性を検証することが可能となる。

#### 【0098】

また、本発明に係るリアクティブ・システムの安全性検証装置において、傍受者の拡張される知識の集合に対応する等式付ツリー・オートマトンの収束性を判断しない場合でも、暗号通信手順が安全でないことを正確に判断することが可能となる。

#### 【0099】

本発明に係るリアクティブ・システムの安全性検証装置によって、一般のリアクティブ・システムに関して、取り得る状態の最大の集合を表す等式付ツリー・オートマトンを生成することができ、検査対象となる状態になり得るか否か、即ちリアクティブ・システムの安全性を検証することが可能となる。

#### 【0100】

大規模システムにおいては、設計の初期段階で安全性の検証をおこなうことができることから、検証によって発見された誤りによって設計変更を余儀なくされた場合でも、損失を小さく抑えることが可能となる。

#### 【0101】

また、本発明に係るリアクティブ・システムの安全性検証装置において、一般的のリアクティブ・システムに関して、拡張される状態の集合を表す等式付ツリー・オートマトンの収束性を判断しない場合でも、リアクティブ・システムが安全でないことを正確に判断することが可能となる。

【図面の簡単な説明】

【図 1】 本発明の実施の形態に係るリアクティブ・システムの安全性検証装置の概略構成を示すブロック図である。

【図 2】 本発明の実施の形態に係るリアクティブ・システムの安全性検証装置が行う処理を示すフローチャートである。

【図 3】 本発明の実施の形態に係るリアクティブ・システムの安全性検証装置による、拡張された知識の集合を受理する等式付ツリー・オートマトンを計算する処理のアルゴリズムを示す説明図である。

【図 4】 2つのオートマトンの結合処理を示す説明図である。

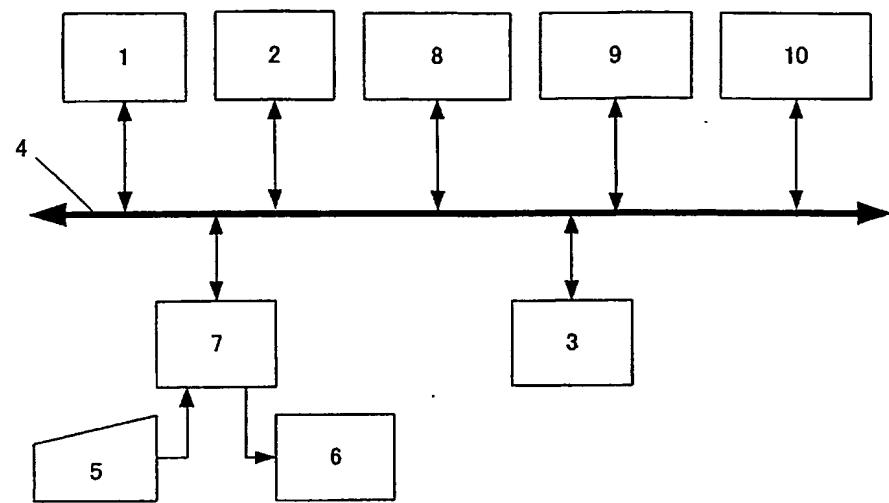
【図 5】 本発明の実施の形態に係るリアクティブ・システムの安全性検証装置が行う処理において、拡張するナレッジを受理する等式付ツリー・オートマトンの収束性を判断しない場合の処理を示すフローチャートである。

【符号の説明】

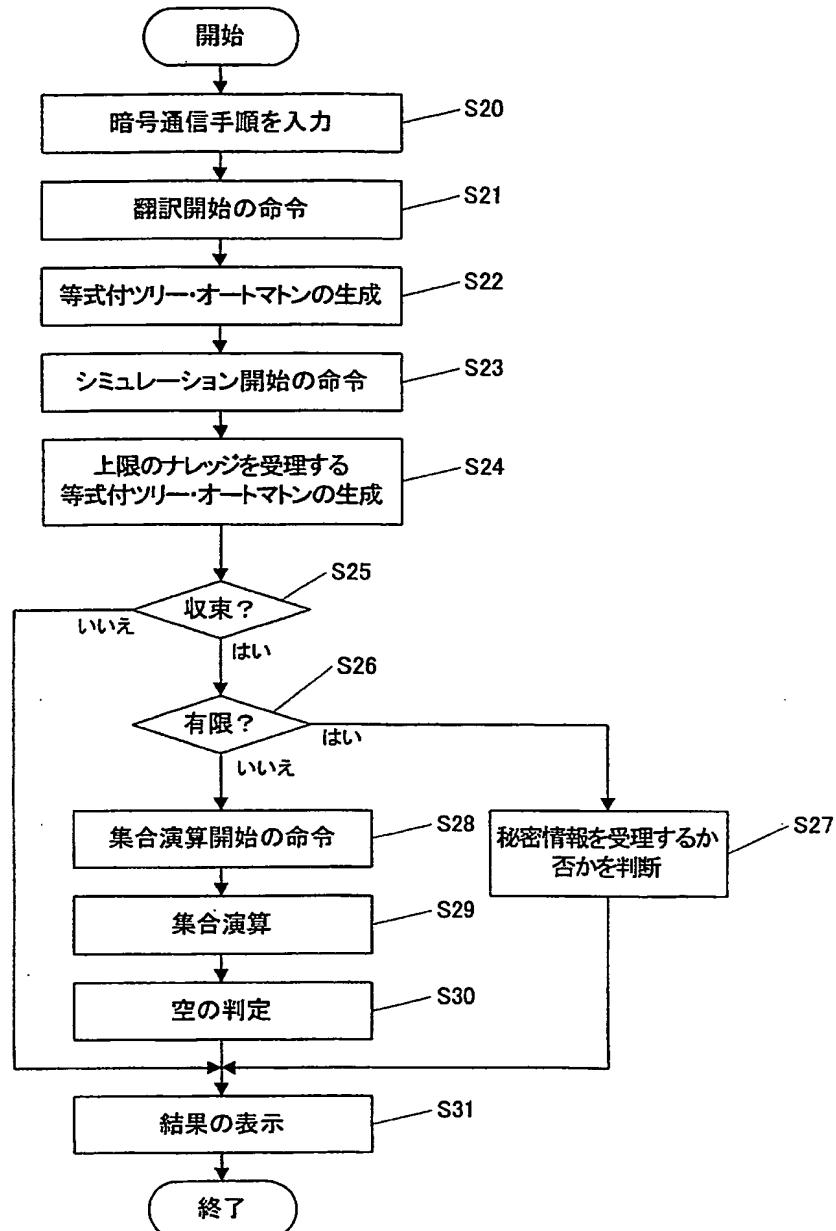
- 1 中央演算処理部 (C P U)
- 2 一時記憶部 (メモリ)
- 3 記録部
- 4 内部バス
- 5 入力部
- 6 表示部
- 7 インタフェース部
- 8 翻訳部
- 9 シミュレーション部
- 10 集合演算部

【書類名】 図面

【図1】



【図 2】



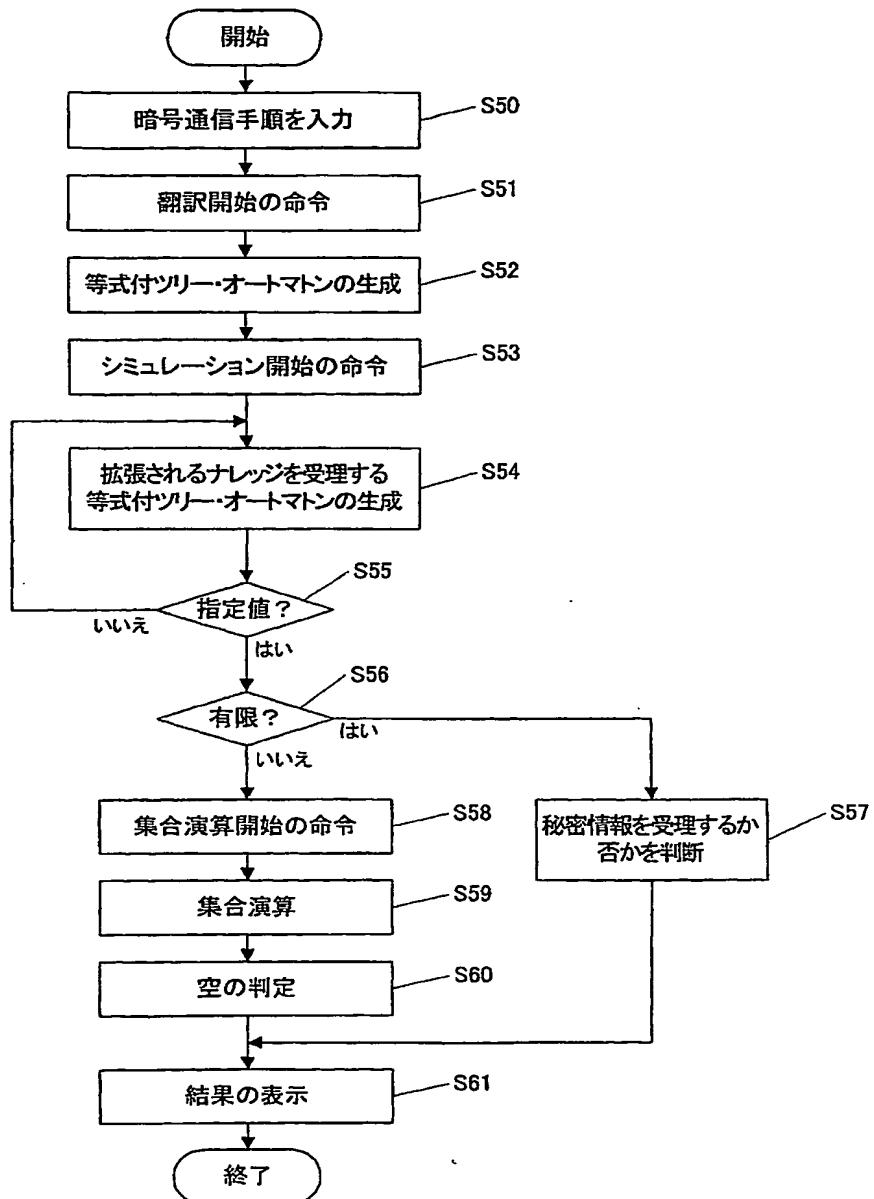
【図 3】

入出力 引数の指定	入力 $l \rightarrow r$ : 書換規則 $A/AC$ : 等式付ツリー・オートマトン 出力 $B_{l \rightarrow r}/AC$ : 等式付ツリー・オートマトン		
初期設定 <table border="0"> <tr> <td style="vertical-align: top; padding-right: 10px;"> <math>A_0 := A</math>; <math>i := 0</math>; <math>j := 0</math>;  <math>S := \text{pos}(l)</math>;  <math>T := \text{pos}(r)</math>;    <b>while</b> <math>S \neq \emptyset</math> <b>do</b> </td> <td>           次の条件を満たす要素 <math>p</math> を <math>S</math> から選択する : <math>\forall p' \in S. p \succeq p'</math>            次の条件を満たす等式付ツリー・オートマトン <math>A_{i+1}/AC</math> を計算する: …(1)  <math>l_{ p} = f(t_1, \dots, t_n)</math> のとき  <math>\mathcal{L}(A_{i+1}/AC) = (\{ \rightarrow_{\{f(c_i^{p,1}, \dots, c_i^{p,n}) \rightarrow c_{i+1}^p\}} / AC) [\mathcal{L}(A_i/AC)]</math> </td> </tr> </table>		$A_0 := A$ ; $i := 0$ ; $j := 0$ ; $S := \text{pos}(l)$ ; $T := \text{pos}(r)$ ;  <b>while</b> $S \neq \emptyset$ <b>do</b>	次の条件を満たす要素 $p$ を $S$ から選択する : $\forall p' \in S. p \succeq p'$ 次の条件を満たす等式付ツリー・オートマトン $A_{i+1}/AC$ を計算する: …(1) $l_{ p} = f(t_1, \dots, t_n)$ のとき $\mathcal{L}(A_{i+1}/AC) = (\{ \rightarrow_{\{f(c_i^{p,1}, \dots, c_i^{p,n}) \rightarrow c_{i+1}^p\}} / AC) [\mathcal{L}(A_i/AC)]$
$A_0 := A$ ; $i := 0$ ; $j := 0$ ; $S := \text{pos}(l)$ ; $T := \text{pos}(r)$ ;  <b>while</b> $S \neq \emptyset$ <b>do</b>	次の条件を満たす要素 $p$ を $S$ から選択する : $\forall p' \in S. p \succeq p'$ 次の条件を満たす等式付ツリー・オートマトン $A_{i+1}/AC$ を計算する: …(1) $l_{ p} = f(t_1, \dots, t_n)$ のとき $\mathcal{L}(A_{i+1}/AC) = (\{ \rightarrow_{\{f(c_i^{p,1}, \dots, c_i^{p,n}) \rightarrow c_{i+1}^p\}} / AC) [\mathcal{L}(A_i/AC)]$		
第1処理 <table border="0"> <tr> <td style="vertical-align: top; padding-right: 10px;"> <math>i := i + 1</math>;  <math>S := S - \{p\}</math>;  <b>od</b> </td> <td>           次の条件を満たす等式付ツリー・オートマトン <math>B_0/AC</math> を計算する  <math>\mathcal{L}(B_0/AC) = (\{ \rightarrow_{\{c_i^p \rightarrow d_i^p\}} / AC) [\mathcal{L}(A_i/AC)]</math> </td> </tr> </table>		$i := i + 1$ ; $S := S - \{p\}$ ; <b>od</b>	次の条件を満たす等式付ツリー・オートマトン $B_0/AC$ を計算する $\mathcal{L}(B_0/AC) = (\{ \rightarrow_{\{c_i^p \rightarrow d_i^p\}} / AC) [\mathcal{L}(A_i/AC)]$
$i := i + 1$ ; $S := S - \{p\}$ ; <b>od</b>	次の条件を満たす等式付ツリー・オートマトン $B_0/AC$ を計算する $\mathcal{L}(B_0/AC) = (\{ \rightarrow_{\{c_i^p \rightarrow d_i^p\}} / AC) [\mathcal{L}(A_i/AC)]$		
第2処理 <table border="0"> <tr> <td style="vertical-align: top; padding-right: 10px;"> <math>j := j + 1</math>;  <math>T := T - \{q\}</math>;  <b>od</b> </td> <td>           次の条件を満たす要素 <math>q</math> を <math>T</math> から選択する : <math>\forall q' \in T. q' \succeq q</math>            次の条件を満たす等式付ツリー・オートマトン <math>B_{j+1}/AC</math> を計算する: …(2)  <math>r_{ q} = f(t_1, \dots, t_n)</math> のとき  <math>\mathcal{L}(B_{j+1}/AC) = (\{ \rightarrow_{\{d_{ q}^p \rightarrow f(d_{i_1}^{q,1}, \dots, d_{i_n}^{q,n})\}} / AC) [\mathcal{L}(B_j/AC)]</math> </td> </tr> </table>		$j := j + 1$ ; $T := T - \{q\}$ ; <b>od</b>	次の条件を満たす要素 $q$ を $T$ から選択する : $\forall q' \in T. q' \succeq q$ 次の条件を満たす等式付ツリー・オートマトン $B_{j+1}/AC$ を計算する: …(2) $r_{ q} = f(t_1, \dots, t_n)$ のとき $\mathcal{L}(B_{j+1}/AC) = (\{ \rightarrow_{\{d_{ q}^p \rightarrow f(d_{i_1}^{q,1}, \dots, d_{i_n}^{q,n})\}} / AC) [\mathcal{L}(B_j/AC)]$
$j := j + 1$ ; $T := T - \{q\}$ ; <b>od</b>	次の条件を満たす要素 $q$ を $T$ から選択する : $\forall q' \in T. q' \succeq q$ 次の条件を満たす等式付ツリー・オートマトン $B_{j+1}/AC$ を計算する: …(2) $r_{ q} = f(t_1, \dots, t_n)$ のとき $\mathcal{L}(B_{j+1}/AC) = (\{ \rightarrow_{\{d_{ q}^p \rightarrow f(d_{i_1}^{q,1}, \dots, d_{i_n}^{q,n})\}} / AC) [\mathcal{L}(B_j/AC)]$		
$B_{l \rightarrow r} := B_j$ ; <b>return</b> $B_{l \rightarrow r}/AC$			

【図 4】

集合	遷移規則	条件
$\mathcal{R}_x$	$f((p_1, q_1), \dots, (p_n, q_n)) \rightarrow (p, q)$	$\forall f \in \mathcal{F} \setminus \mathcal{G}$ $\forall f(p_1, \dots, p_n) \rightarrow p \in \mathcal{R}_A$ $\forall f(q_1, \dots, q_n) \rightarrow q \in \mathcal{R}_B$
$\mathcal{R}_{\overline{A}}$	$g((p_1, q_1), (p_2, q_2)) \rightarrow g((p, q_1), q_2)$ $g(p_1, (p_2, q_2)) \rightarrow (p, q_2)$	$\forall g \in \mathcal{G}$ $\forall q_1, q_2 \in \mathcal{Q}_B$ $\forall g(p_1, p_2) \rightarrow p \in \mathcal{R}_A$
	$g((p_1, q_1), (p_2, q_2)) \rightarrow g((r_1, q_1), (r_2, q_2))$ $g(p_1, (p_2, q_2)) \rightarrow g(r_1, (r_2, q_2))$	$\forall g(p_1, p_2) \rightarrow g(r_1, r_2) \in \mathcal{R}_A$
$\mathcal{R}_{\overline{B}}$	$g((p_1, q_1), (p_2, q_2)) \rightarrow g((p_1, q), p_2)$ $g(q_1, (p_2, q_2)) \rightarrow (p_2, q)$	$\forall g \in \mathcal{G}$ $\forall p_1, p_2 \in \mathcal{Q}_A$ $\forall g(q_1, q_2) \rightarrow q \in \mathcal{R}_B$
	$g((p_1, q_1), (p_2, q_2)) \rightarrow g((p_1, r_1), (p_2, r_2))$ $g(q_1, (p_2, q_2)) \rightarrow g(r_1, (p_2, r_2))$	$\forall g(q_1, q_2) \rightarrow g(r_1, r_2) \in \mathcal{R}_B$
$\mathcal{R}_g$	$g((p, q_1), q_2) \rightarrow g(q_1, (p, q_2))$ $g((p_1, q), p_2) \rightarrow g(p_1, (p_2, q))$ $g(q, p) \rightarrow (p, q)$	$\forall g \in \mathcal{G}$ $\forall p_1, p_2, p \in \mathcal{Q}_A$ $\forall q_1, q_2, q \in \mathcal{Q}_B$

【図5】



【書類名】 要約書

【要約】

【課題】 等式付ツリー・オートマトン理論の範疇にあるリアクティブ・システムの動作手順の安全性を検証すること。

【解決手段】 リアクティブ・システムの安全性検証装置において、公理の集合が交換則及び結合則のみを要素とし、前記公理の集合の下で、項の集合を受理する第1の等式付ツリー・オートマトンを生成する翻訳部8、前記第1の等式付ツリー・オートマトンを初期データとして、書換規則の集合及び前記公理の集合の下で、前記項の集合から派生する項からなる集合と前記項の集合とを受理する第2の等式付ツリー・オートマトンを生成するシミュレーション部9、及び前記第2の等式付ツリー・オートマトンと、検査対象となる項の集合を受理する第3の等式付ツリー・オートマトンとを結合して第4の等式付ツリー・オートマトンを生成し、該第4の等式付ツリー・オートマトンが受理する集合が空集合か否かを判断する集合演算部10を備えている。

【選択図】 図1

特願 2002-211021

出願人履歴情報

識別番号 [301021533]

1. 変更年月日 2001年 4月 2日

[変更理由] 新規登録

住 所 東京都千代田区霞が関 1-3-1  
氏 名 独立行政法人産業技術総合研究所

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- BLACK BORDERS**
- IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- FADED TEXT OR DRAWING**
- BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- SKEWED/SLANTED IMAGES**
- COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- GRAY SCALE DOCUMENTS**
- LINES OR MARKS ON ORIGINAL DOCUMENT**
- REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**